## jsonp解决的问题

ajax不能跨域

## jsonp的原理

script标签可以跨域

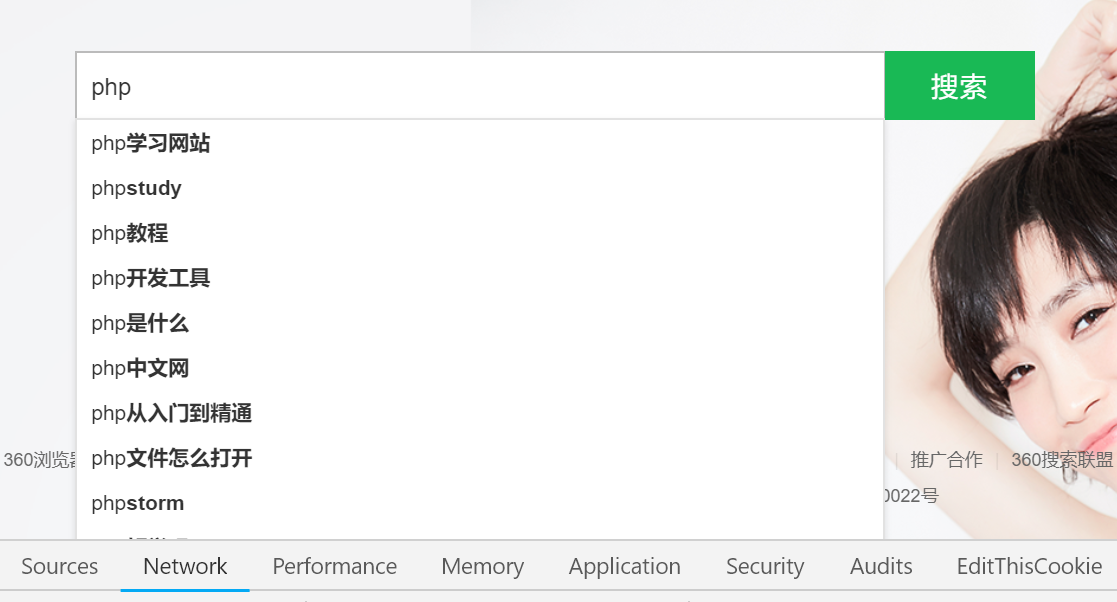
<https://cdn.bootcss.com/jquery/1.12.4/jquery.js>

比如我们可以写一个跨域的js文件，这个文件当中写的是一个函数的调用，我们在本地提前创建好这个函数，这是被允许的

既然我们的php使用json\_encode的数据并不能使用ajax返回给我们，我们就干脆把这个json\_encode的数据当成一个实参，搞一个函数名包着，然后在本地提前创建好一个函数，这个函数的形参对应的就是json\_encode的数据，那这样的话，我们不就可以在函数体内部得到这个数据吗？

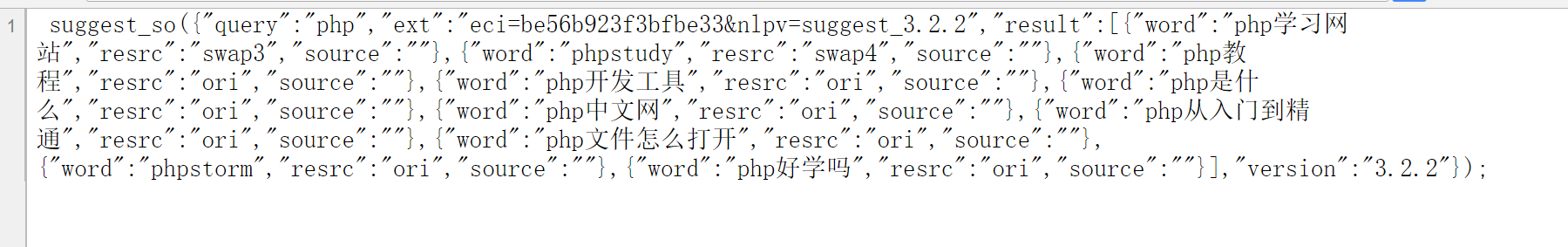
如果上面这段话你能够理解，那就基本上理解了jsonp的原理

## 使用jsonp完成类似so.com一样的效果

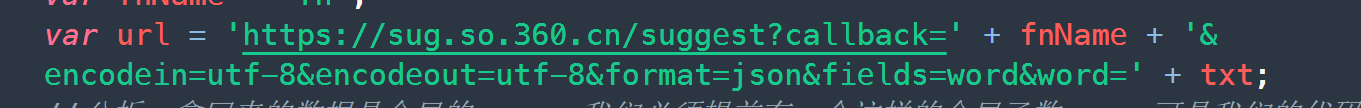


<https://sug.so.360.cn/suggest?callback=suggest_so&encodein=utf-8&encodeout=utf-8&format=json&fields=word&word=php>

观察数据可以得知，so.com这个查询猜想词拿到的数据是跨域了，并且拿回来的数据和我们之前使用ajax不太一样，得到的数据是一个函数



说明使用办法：



必须指定一个fn函数名和word查询关键字

## 封装jsonp

我们封装的代码是基于上面的搜索案例

### 第一版

思考

把所有的涉及到jsonp的代码全部提取到函数jsonp当中（数据需要传入，所以需要有一个参数）

|  |
| --- |
| function jsonp(data){  var fnName = "fn";  var url = 'https://sug.so.360.cn/suggest?callback=' + fnName + '&encodein=utf-8&encodeout=utf-8&format=json&fields=word&word=' + data;  //分析：拿回来的数据是全局的 ---> 我们必须提前有一个这样的全局函数 --> 可是我们的代码已经写到这里来了，我们怎么样才能创建一个全局的函数呢?  window[fnName] = function (data) {  console.log(data);  var html = template('searchTpl', data);  document.querySelector('.so\_result').innerHTML = html;  document.querySelector('.so\_result').style.display = 'block';  }  var oScript = document.createElement('script');  oScript.setAttribute("src", url);  document.body.appendChild(oScript);  }  window.onload = function () {  var search\_content = document.querySelector('#search\_content');  search\_content.onkeyup = function () {  console.log(this.value);  var txt = this.value;  jsonp(txt);  }  search\_content.onblur = function(){  document.querySelector('.so\_result').style.display = 'none';  }  } |

### 第二版

思考：

1. 如果传入一堆的数据作为url的参数，根据我们之前封装$.ajax的经验，有两种思路
   1. a=1&b=2
   2. {a:1,b:2}
2. 思路
   1. 参数用对象来表表示
   2. data数据可以让用户传入a=1&b=2或者{a:1,b:2}的形式

|  |
| --- |
| //data可以传入的形式 'a=1&b=2'或者{a:1,b:2}  function jsonp(option){  if(typeof option.data == 'object'){  var str = '';  for(var i in option.data){  str += i + '=' + option.data[i] + '&';  }  option.data = str.slice(0,str.length-1);  }  var fnName = "fn";  var url = 'https://sug.so.360.cn/suggest?callback=' + fnName + '&' + option.data;  console.log(url);  //分析：拿回来的数据是全局的 ---> 我们必须提前有一个这样的全局函数 --> 可是我们的代码已经写到这里来了，我们怎么样才能创建一个全局的函数呢?  window[fnName] = function (data) {  console.log(data);  var html = template('searchTpl', data);  document.querySelector('.so\_result').innerHTML = html;  document.querySelector('.so\_result').style.display = 'block';  }  var oScript = document.createElement('script');  oScript.setAttribute("src", url);  document.body.appendChild(oScript);  }  window.onload = function () {  var search\_content = document.querySelector('#search\_content');  search\_content.onkeyup = function () {  console.log(this.value);  var txt = this.value;  jsonp({  data:{  encodein:'utf-8',  encodeout:'utf-8',  format:'json',  fields:'word',  word:txt  }  });  }  search\_content.onblur = function(){  document.querySelector('.so\_result').style.display = 'none';  }  } |

### 第三版

把url也做为参数传入

|  |
| --- |
| //data可以传入的形式 'a=1&b=2'或者{a:1,b:2}  function jsonp(option){  if(typeof option.data == 'object'){  var str = '';  for(var i in option.data){  str += i + '=' + option.data[i] + '&';  }  option.data = str.slice(0,str.length-1);  }  var fnName = "fn";  var url = option.url + '?callback=' + fnName + '&' + option.data;  console.log(url);  //分析：拿回来的数据是全局的 ---> 我们必须提前有一个这样的全局函数 --> 可是我们的代码已经写到这里来了，我们怎么样才能创建一个全局的函数呢?  window[fnName] = function (data) {  console.log(data);  var html = template('searchTpl', data);  document.querySelector('.so\_result').innerHTML = html;  document.querySelector('.so\_result').style.display = 'block';  }  var oScript = document.createElement('script');  oScript.setAttribute("src", url);  document.body.appendChild(oScript);  }  window.onload = function () {  var search\_content = document.querySelector('#search\_content');  search\_content.onkeyup = function () {  console.log(this.value);  var txt = this.value;  jsonp({  url:'https://sug.so.360.cn/suggest',  data:{  encodein:'utf-8',  encodeout:'utf-8',  format:'json',  fields:'word',  word:txt  }  });  }  search\_content.onblur = function(){  document.querySelector('.so\_result').style.display = 'none';  }  } |

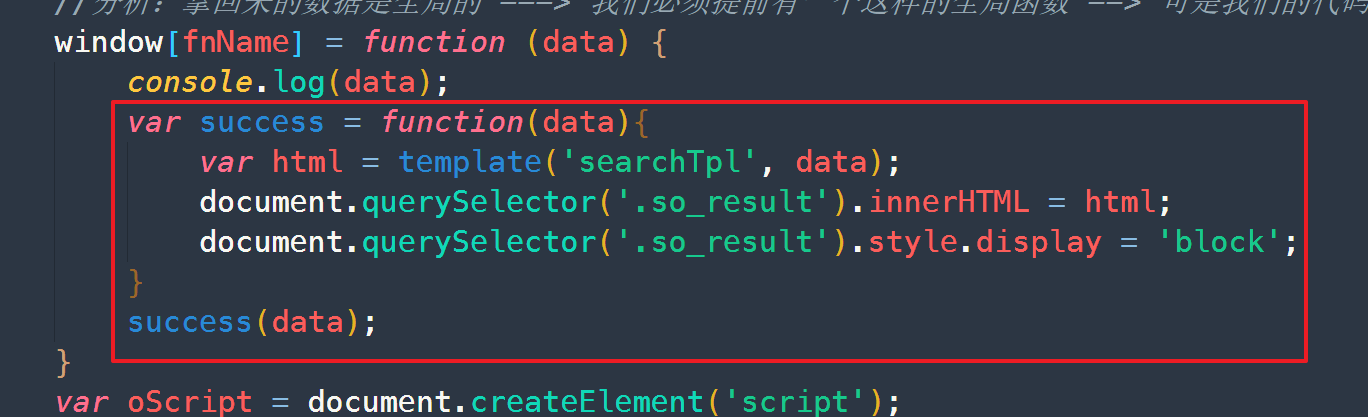
### 第四版

把拿到数据之后做的事情提取成一个函数

|  |
| --- |
| //data可以传入的形式 'a=1&b=2'或者{a:1,b:2}  function jsonp(option){  if(typeof option.data == 'object'){  var str = '';  for(var i in option.data){  str += i + '=' + option.data[i] + '&';  }  option.data = str.slice(0,str.length-1);  }  var fnName = "fn";  var url = option.url + '?callback=' + fnName + '&' + option.data;  console.log(url);  //分析：拿回来的数据是全局的 ---> 我们必须提前有一个这样的全局函数 --> 可是我们的代码已经写到这里来了，我们怎么样才能创建一个全局的函数呢?  window[fnName] = function (data) {  console.log(data);  function success(data){  var html = template('searchTpl', data);  document.querySelector('.so\_result').innerHTML = html;  document.querySelector('.so\_result').style.display = 'block';  }  success(data);  }  var oScript = document.createElement('script');  oScript.setAttribute("src", url);  document.body.appendChild(oScript);  }  window.onload = function () {  var search\_content = document.querySelector('#search\_content');  search\_content.onkeyup = function () {  console.log(this.value);  var txt = this.value;  jsonp({  url:'https://sug.so.360.cn/suggest',  data:{  encodein:'utf-8',  encodeout:'utf-8',  format:'json',  fields:'word',  word:txt  }  });  }  search\_content.onblur = function(){  document.querySelector('.so\_result').style.display = 'none';  }  } |

### 第五版

改造函数的表示形式



### 第六版

把success作为参数提取出来

|  |
| --- |
| //data可以传入的形式 'a=1&b=2'或者{a:1,b:2}  function jsonp(option){  if(typeof option.data == 'object'){  var str = '';  for(var i in option.data){  str += i + '=' + option.data[i] + '&';  }  option.data = str.slice(0,str.length-1);  }  var fnName = "fn";  var url = option.url + '?callback=' + fnName + '&' + option.data;  console.log(url);  //分析：拿回来的数据是全局的 ---> 我们必须提前有一个这样的全局函数 --> 可是我们的代码已经写到这里来了，我们怎么样才能创建一个全局的函数呢?  window[fnName] = function (data) {  console.log(data);  option.success(data);  }  var oScript = document.createElement('script');  oScript.setAttribute("src", url);  document.body.appendChild(oScript);  }  window.onload = function () {  var search\_content = document.querySelector('#search\_content');  search\_content.onkeyup = function () {  console.log(this.value);  var txt = this.value;  var success = function (data) {  var html = template('searchTpl', data);  document.querySelector('.so\_result').innerHTML = html;  document.querySelector('.so\_result').style.display = 'block';  }  jsonp({  url:'https://sug.so.360.cn/suggest',  data:{  encodein:'utf-8',  encodeout:'utf-8',  format:'json',  fields:'word',  word:txt  },  success:success  });  }  search\_content.onblur = function(){  document.querySelector('.so\_result').style.display = 'none';  }  } |

### 第七版

测试jsonp方法（调用两次）



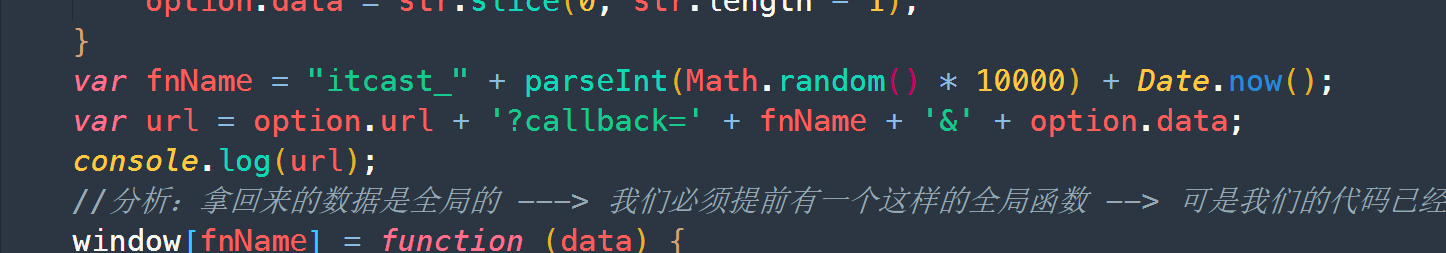
结果：打印了两次2222

思考为什么：

1. 两次调用冲突了
2. 我们二次调用的函数名都叫fn,都是全局的，后面的把全面的覆盖掉了

解决办法

### 第八版



第九版

把jsonp方法集成到$.ajax当中